Report on the Investigation

Word count: 2788

1 Comparative study on Part 2

1.1 Brief description of the methods' implementation

Since the next step includes finding the best model with optimal parameters using GridSearchCV, there were made up sets for each classifier with various parameters to test that are reflected in the tables below.

Decision trees

Table 1 – The set of various parameters for the Decision tree method.

The name of a parameter	The range of a parameter	The purpose of a parameter
criterion	"gini","entropy"	Tree branching quality estimation
max_depth	1-29 ,"None"	Specification of a decision tree's maximum depth
min_samples_split	2-7	Specification of a minimum number of samples in the tree to split
min_samples_leaf	1-15	Specification of the minimum number of tree leaves

K-nearest neighbours

Table 2 – The set of various parameters for the K-nearest neighbours' method.

The name of a parameter	The range of a parameter	The purpose of a parameter
n_neighbors	1-150	K- number of the most similar samples to the considered one

Support vector machine

Table 3 – The set of various parameters for the Support vector machine method with a radial basis function kernel (SVM with rbf).

The name of a parameter	The range of a parameter	The purpose of a parameter
gamma	0.0001-0.050	A specified coefficient for the kernel

Table 4 – The set of various parameters for the SVM with a linear function kernel.

The name of a parameter	The range of a parameter	The purpose of a parameter
C	0.1-10.0	The regularization parameter of the SVM

Besides these sets, *GridSearchCV* for each model is initialized with several other main parameters such as:

- *cv = 5* – a number of cross-validation folds
- *scoring = "accuracy"* – scoring a model measuring a classification accuracy
- *n_jobs = -1* – specification of a maximum number of available CPUs to speed up finding an optimal model

1.2 Obtained results

The best parameters for each model, that were revealed by GridSearchCV, can be seen in the table below.

Table 5 – The best parameters for each model

Parameter name	Parameter value
K-nearest neighbours	
n_neighbors	43
SVM with rbf	
gamma	0.0097
SVM with a linear function kernel	
C	9.9
Decision trees classifier	
criterion	'gini'
max_depth	9
min_samples_split	2
min_samples_leaf	9

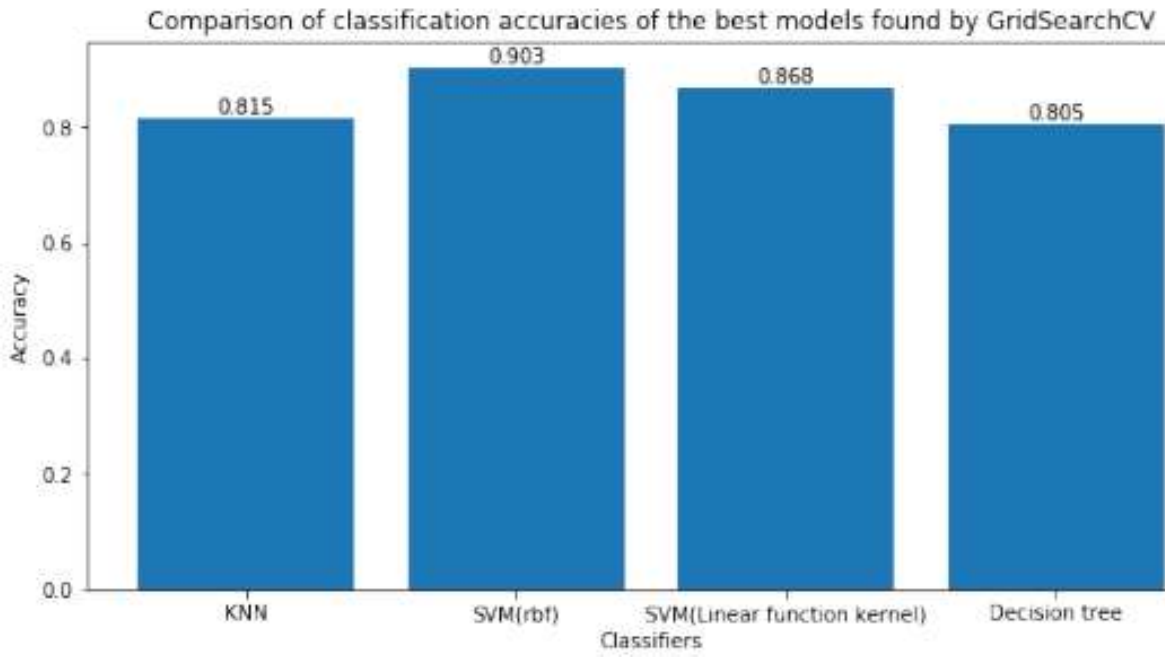


Figure 1 – A bar chart of mean classification accuracy comparison among four methods after performed cross-validation in the GridSearchCV.

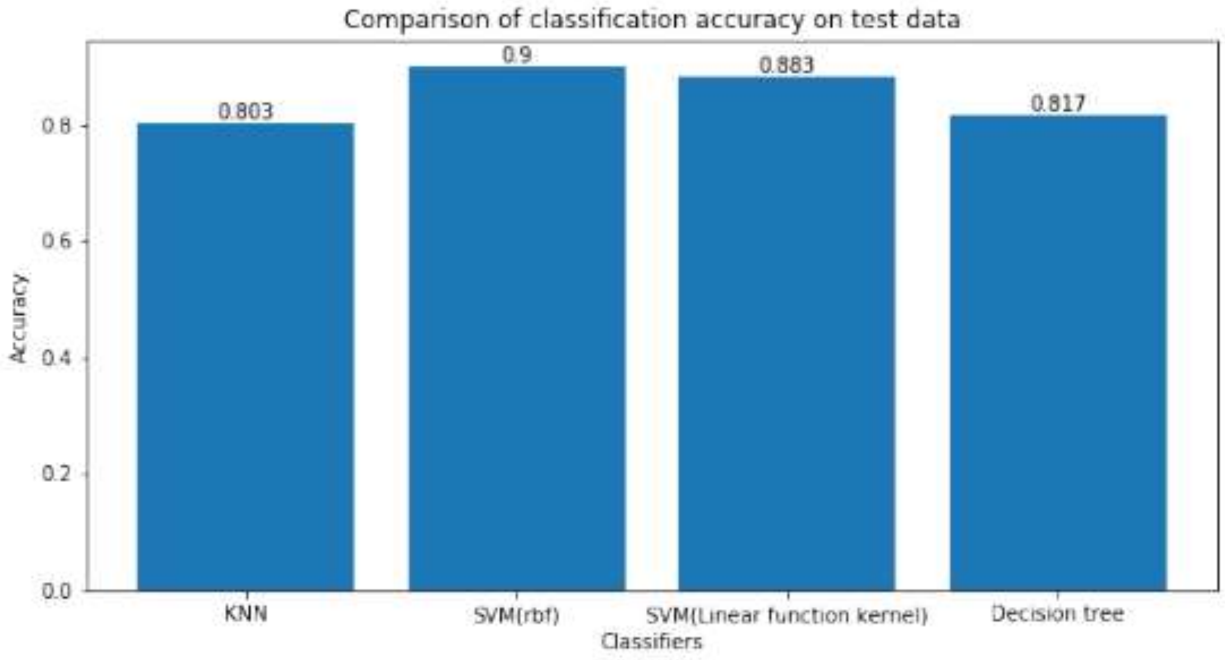
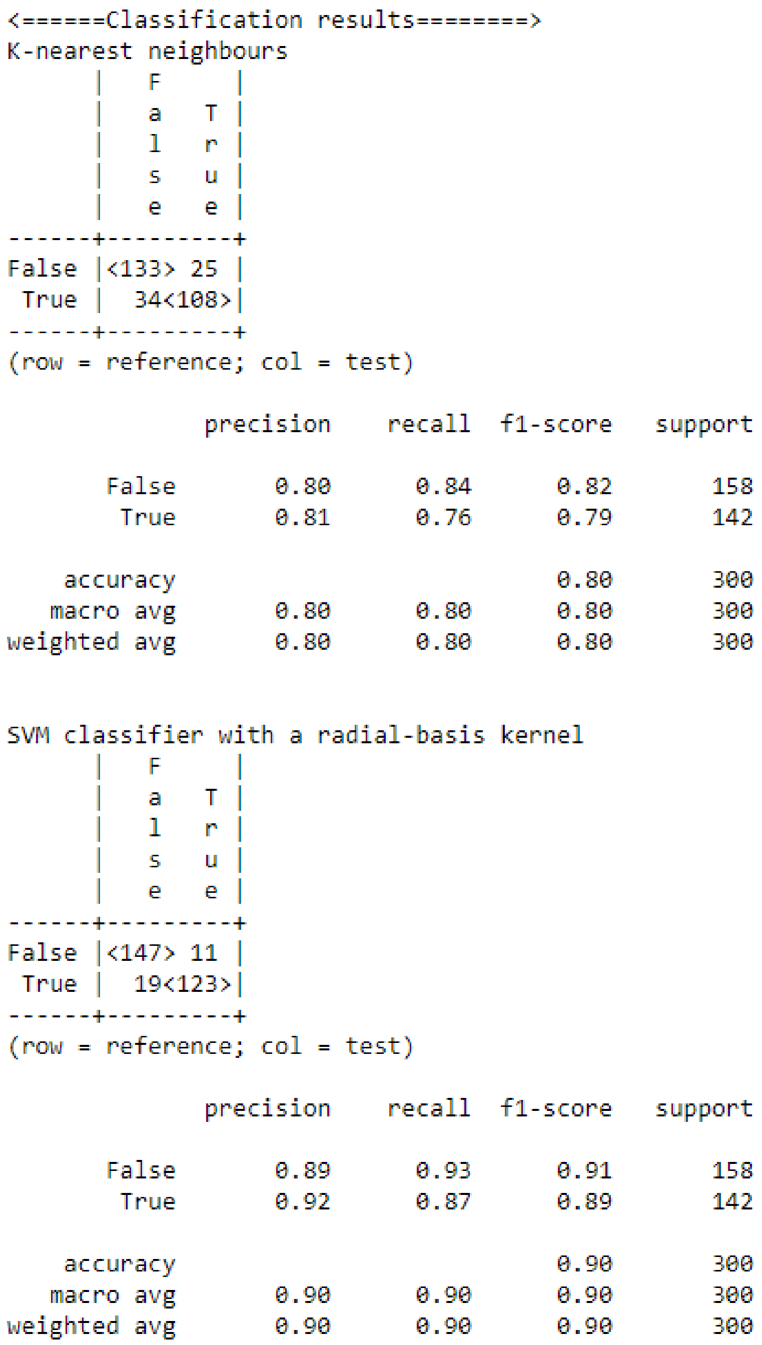


Figure 2 – A bar chart of mean classification accuracy comparison among three methods on the prepared test share of the training set.

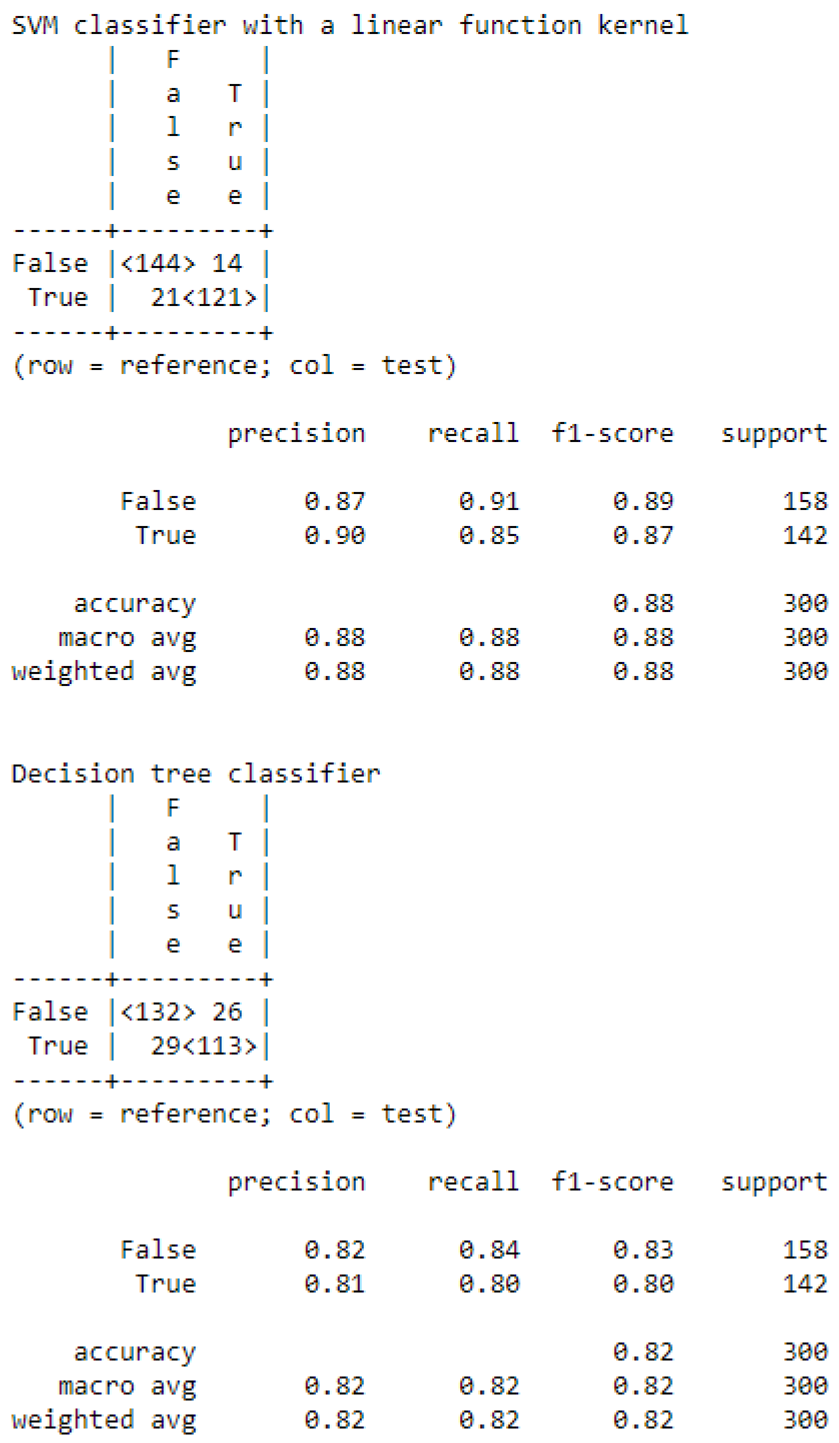
According to figures 1,2 above, it can be noticed that the largest recession of classification accuracy is inherent in k-nearest neighbours, while the other classifiers' accuracies have increased their ones but by a small extent. Besides that, the SVM with rbf model has reached the highest accuracy, while the k-nearest neighbours model– the lowest one.

Apart from the accuracies results on figures 3,4 below there is reflected more information about other metrics such as precision, recall, f1-score retrieved in the result of the prediction.



```
<======Classification results=======>
K-nearest neighbours
      |   F     |
      |   a   T |
      |   l   r |
      |   s   u |
      |   e   e |
------+---------+
False |<133> 25 |
 True |   34<108>|
------+---------+
(row = reference; col = test)

              precision    recall  f1-score   support

       False       0.80      0.84      0.82       158
        True       0.81      0.76      0.79       142

    accuracy                           0.80       300
   macro avg       0.80      0.80      0.80       300
weighted avg       0.80      0.80      0.80       300


SVM classifier with a radial-basis kernel
      |   F     |
      |   a   T |
      |   l   r |
      |   s   u |
      |   e   e |
------+---------+
False |<147> 11 |
 True |   19<123>|
------+---------+
(row = reference; col = test)

              precision    recall  f1-score   support

       False       0.89      0.93      0.91       158
        True       0.92      0.87      0.89       142

    accuracy                           0.90       300
   macro avg       0.90      0.90      0.90       300
weighted avg       0.90      0.90      0.90       300
```

Figure 3 – A detailed report on prediction performance for K-NN and SVM with rbf.



```
SVM classifier with a linear function kernel
      |    F     |
      |    a   T |
      |    l   r |
      |    s   u |
      |    e   e |
------+----------+
False |<144> 14  |
 True |   21<121>|
------+----------+
(row = reference; col = test)

              precision    recall  f1-score   support

       False       0.87      0.91      0.89       158
        True       0.90      0.85      0.87       142

    accuracy                           0.88       300
   macro avg       0.88      0.88      0.88       300
weighted avg       0.88      0.88      0.88       300


Decision tree classifier
      |    F     |
      |    a   T |
      |    l   r |
      |    s   u |
      |    e   e |
------+----------+
False |<132> 26  |
 True |   29<113>|
------+----------+
(row = reference; col = test)

              precision    recall  f1-score   support

       False       0.82      0.84      0.83       158
        True       0.81      0.80      0.80       142

    accuracy                           0.82       300
   macro avg       0.82      0.82      0.82       300
weighted avg       0.82      0.82      0.82       300
```

Figure 4 – A detailed report on prediction performance for K-NN and SVM with rbf.

As it can be clearly seen from figures 3,4 above that a share of correct predictions is the most significant in the confusion matrix of the SVM classifier with rbf.

Eventually, as the result of the prediction of target values from the test dataset by trained classification models, the following result table was made up. This table is depicted below.

Table 6 – Comparative table of results obtained from applied classification models to the test dataset.

Predicted #	Predicted values			
	K-nearest neighbours	Support vector machine with a linear function kernel	Support vector machine with rbf	Decision trees
0	False	True	False	False
1	True	False	True	True
2	**True**	**True**	**True**	**True**
3	**False**	**False**	**False**	**False**
4	**False**	**False**	**False**	**False**
5	**False**	**False**	**False**	**False**
6	**False**	**False**	**False**	**False**
7	**True**	**True**	**True**	**True**
8	True	True	True	False
9	**True**	**True**	**True**	**True**
10	**True**	**True**	**True**	**True**
11	**True**	**True**	**True**	**True**
12	**False**	**False**	**False**	**False**

While considering results from the sample above we can notice one interesting detail regarding the content of table 6. In essence, the table contains two clusters of identical results from each of the methods. Thus, this feature indicates the consistency between the forecast results of the same data and that those classifiers have very good performance. Besides that, it became clear that those classifiers can determine whether a customer is likely to get insured for the next time or not.

1.3 Summary

According to obtained results from the conducted investigation, we should state that we have managed to determine whether a certain customer shall obtain a discounted premium and get insured for the next time or not by applying the ML techniques.

The list of implemented techniques includes a decision tree classifier, a support vector machine with a radial basis function kernel, a support vector machine with a linear function kernel, and a k-nearest neighbours' classifier. Thus, by applying unique features of each method to solve the given task, all of these classifiers have provided rather fine performance both on train and test datasets.

By the way, we can determine the strengths and weaknesses of applied methods by considering the obtained results.

For example, the main strength of the decision tree is its powerful ability to find hidden rules from the given data and represent them as a tree. It allows anyone to better understand what are the logical connections between different variables. On the other hand, in our case, the obtained tree was too large, so it turned out hard to perceive anything. Besides that, this method is not powerful enough to achieve high accuracy on rather large volumes of real data. In addition, it has too many parameters to adjust, so it could be a hard task to find the optimal ones to reach a decent level of precision.

In contrast, both SVM implementations have a provided decent performance on such large datasets, but the best one had a radial basis kernel which allowed it to achieve 90% of accuracy on the test data. Therefore, the main strength is their excellent performance and applicability to a vast amount of information. However, the weakness of SVM is its sensibility to a scale of data. This fact indicates that a researcher has to investigate how to standardise his raw data in order to obtain a precise model.

The last model – k-nearest neighbours has got a similar accuracy to the decision tree one. In essence, it also has the same flaw as the decision tree method – sensibility to a volume of data, so it does not perform enough well on large datasets.

On the other hand, it does not have many parameters to adjust. So, due to the basic idea embedded in its algorithm, it has proven its applicability to the given datasets and demonstrated a decent performance indeed.

All in all, in the result of the conducted investigation on the ML methods application to classify customers, it should be claimed that the given task can be solved using such classifiers as decision trees, support vector machine with a radial basis function kernel, a support vector machine with a linear function kernel and a k-nearest neighbours. Thus, in essence, a travel insurance company can apply the aforementioned techniques to get to know whether they should suggest a discounted premium further or not.

2 Comparative study on Part 3

2.1 Brief description of implemented methods

Since the next step includes finding the best model with optimal parameters using GridSearchCV, there were made up sets for each method with various parameters to test that are reflected in tables 7,8 below.

Support vector machine

Table 7 – The set of various parameters for the Support vector machine method with a radial basis function kernel.

The name of a parameter	The range of a parameter	The purpose of a parameter
C	0.1-5.0	The regularization parameter of the SVM

Multi-layer perceptron

Table 8 – The set of various parameters for the Multi-layer perceptron method.

The name of a parameter	The range of a parameter	The purpose of a parameter
hidden_layer_sizes	(50), (50,50), (50,50,50), (50,50,50,50), (50,50,50,50,50), (50,50,50,50,50,50), (50,50,50,50,50,50,50)	Content description of all layers in the neural network model

Linear regression

Since this method does not have any specific hyperparameters to adjust, it does not need the GridSearchCV step either. Instead, it only uses cross_val_score method for performing cross-validation and estimates a mean squared error of prediction on the test data.

Besides those sets, Cross_val_score and *GridSearchCV* for each model are initialized with several other main parameters such as:

- *cv = 5* – a number of cross-validation folds
- *scoring = "neg_mean_squared_error"* – scoring a model measuring a classification accuracy
- *n_jobs = -1* – specification of a maximum number of available CPUs to speed up finding an optimal model

2.2 Obtained results

The best parameters for each model, that were revealed by GridSearchCV, can be seen in the table below.

Table 9 – The best parameters for each model

Parameter name	Parameter value
Multi-layer perceptron	
hidden_layer_sizes	(50, 50)
Support vector machine	
C	2.6

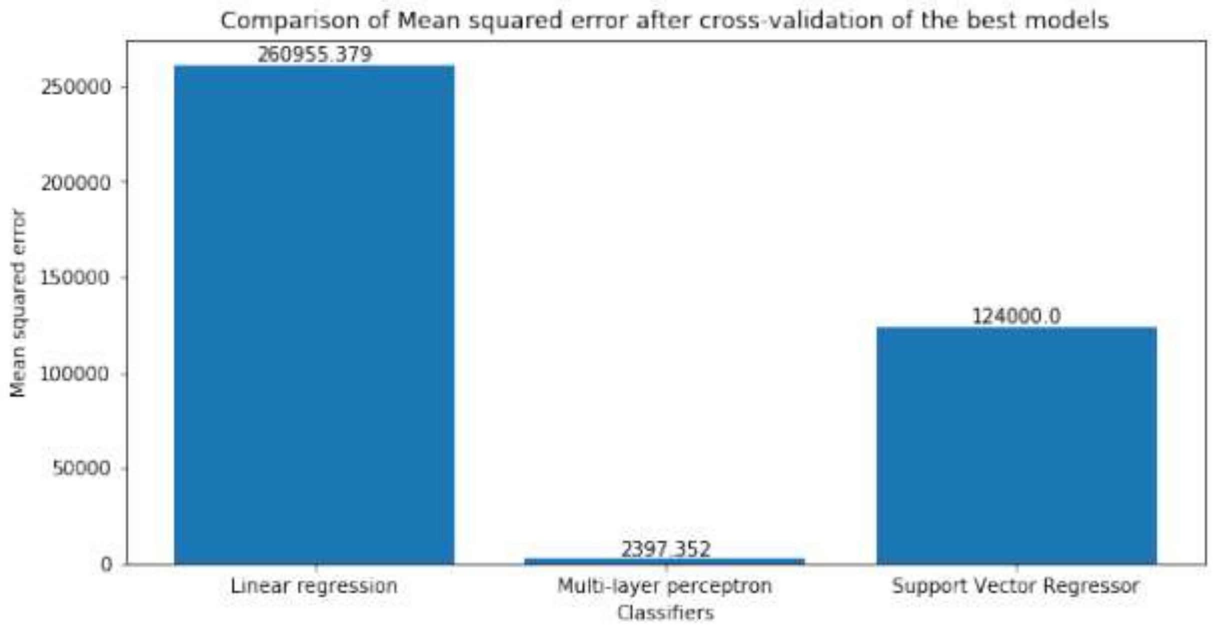


Figure 5 – A bar chart of the mean squared error comparison among three methods after cross-validation in the GridSearchCV.

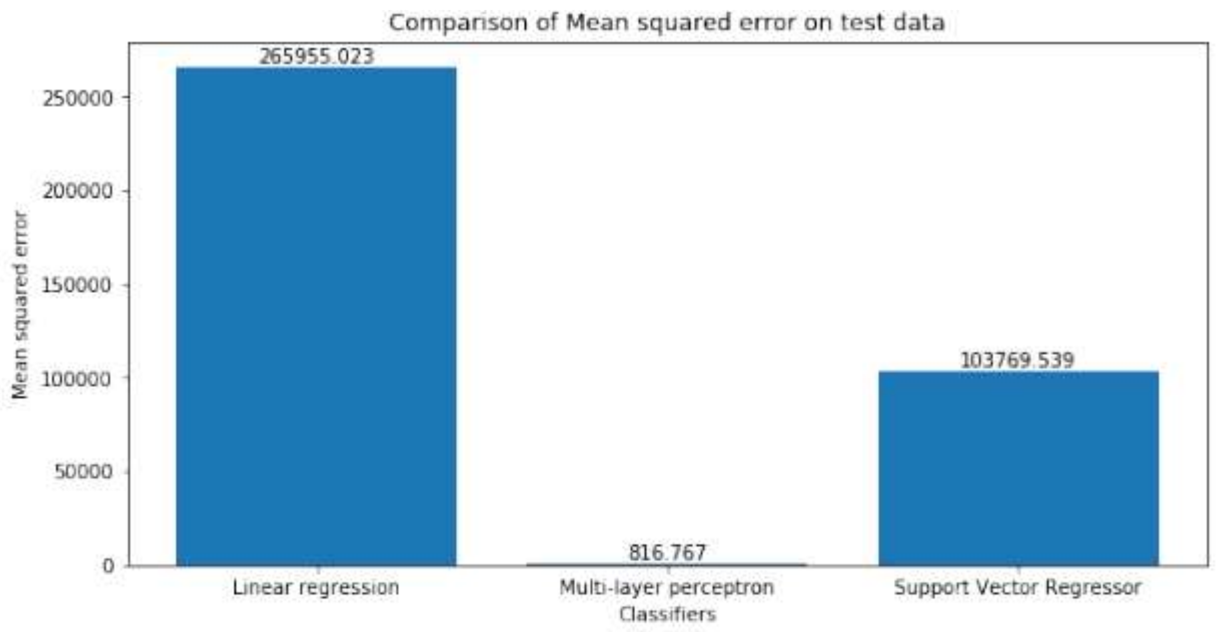


Figure 6 – A bar chart of the mean squared error comparison among three methods on the prepared test share of the training set.

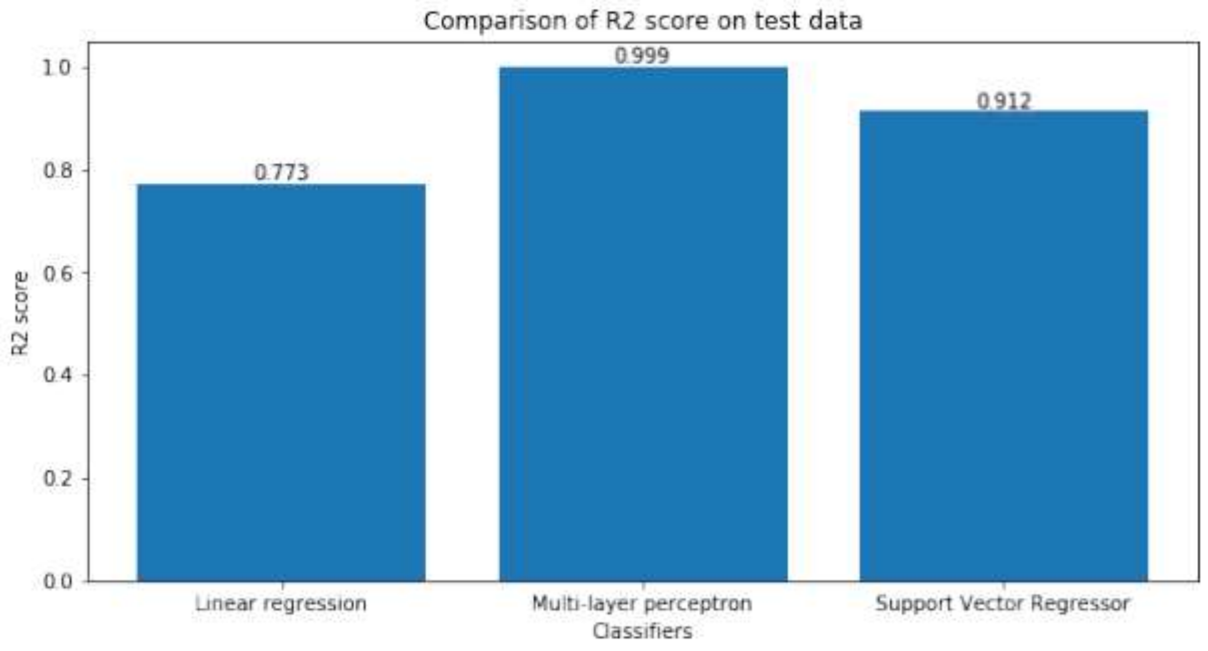


Figure 7 – A bar chart of the R2 score comparison among three methods on the prepared test share of the training set.

Considering the results reflected in figures 5,6,7 above, it should be stated that even after testing the multi-layer perceptron model on the unseen data its mean squared error has decreased. Moreover, the R2 score on the test data is almost one. Thus, both of these facts confirm the assumption that the model was trained perfectly and therefore has a relatively tiny prediction error.

The difference between the accuracies of each method is demonstrated better in table 10 below.

Table 10 – The table of comparison between actual values and predicted ones.

Record	Actual	Predicted values		
		Multi-layer perceptron	Support vector machine	Linear regression
0	475.90	508.539091	588.486319	726.400002
1	171.50	158.726110	183.146305	385.820427
2	728.66	784.578626	913.438289	956.316462
3	0.00	1.177026	-43.762342	-344.788691
4	580.16	589.585834	776.467087	906.901620
5	0.00	-2.733381	253.393827	516.467786
6	0.00	-1.826787	-33.665007	-425.026423
7	2031.42	1960.445351	1764.234917	1761.464708
8	0.00	-18.061600	-107.127773	-162.947552
9	0.00	0.184409	533.353645	704.834123
10	2908.23	2855.745356	3967.994901	2647.761125

While comparing the actual values and predicted ones, it can be noticed that the Multi-layer perceptron model has almost identical results, which indicates its reliability and indeed excellent performance.

In contrast to it, the Linear regression model has shown the worst prediction results that are rather distant from the real values.

By the way, it is worth mentioning a fact that SVM also finely managed with the task, because in most cases its forecasted values are rather similar to the actual ones, despite having sometimes a large bias.

Besides the table above, the significance of squared errors for each prediction result is reflected in figures 8,9,10 below.

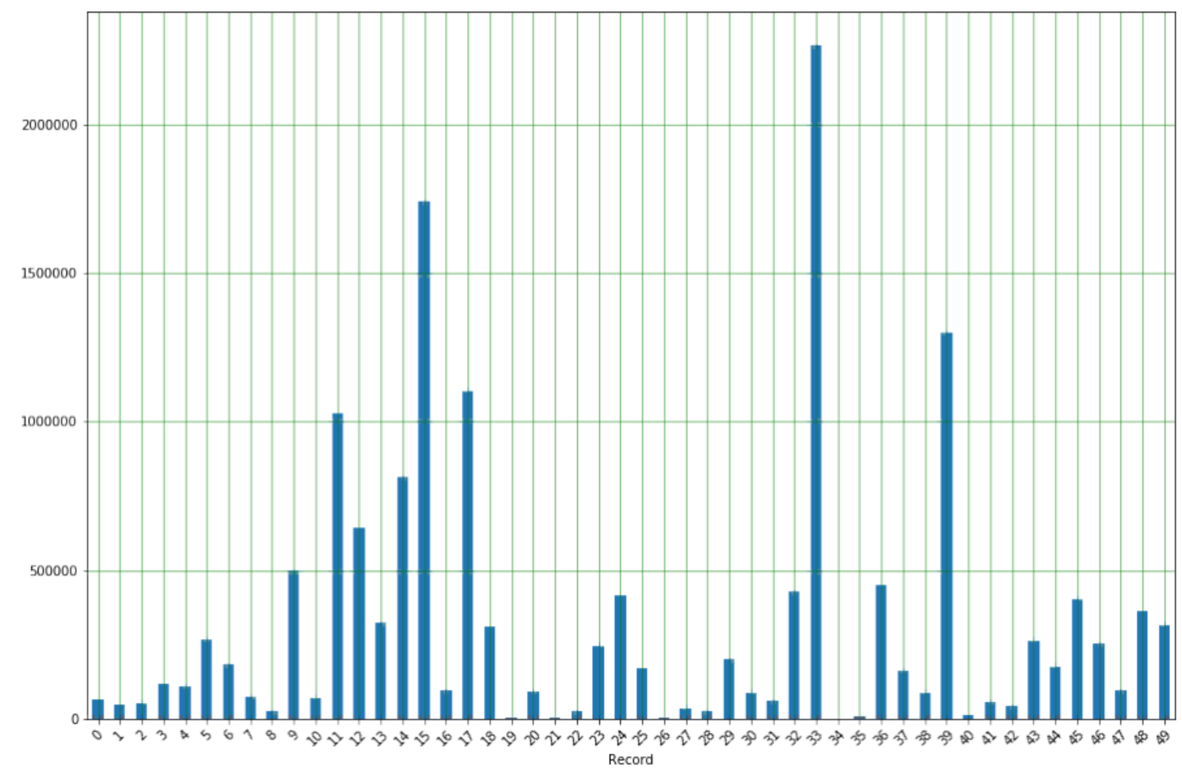


Figure 8 – A bar chart of the first 50 squared errors of the linear regression model for each prediction result.

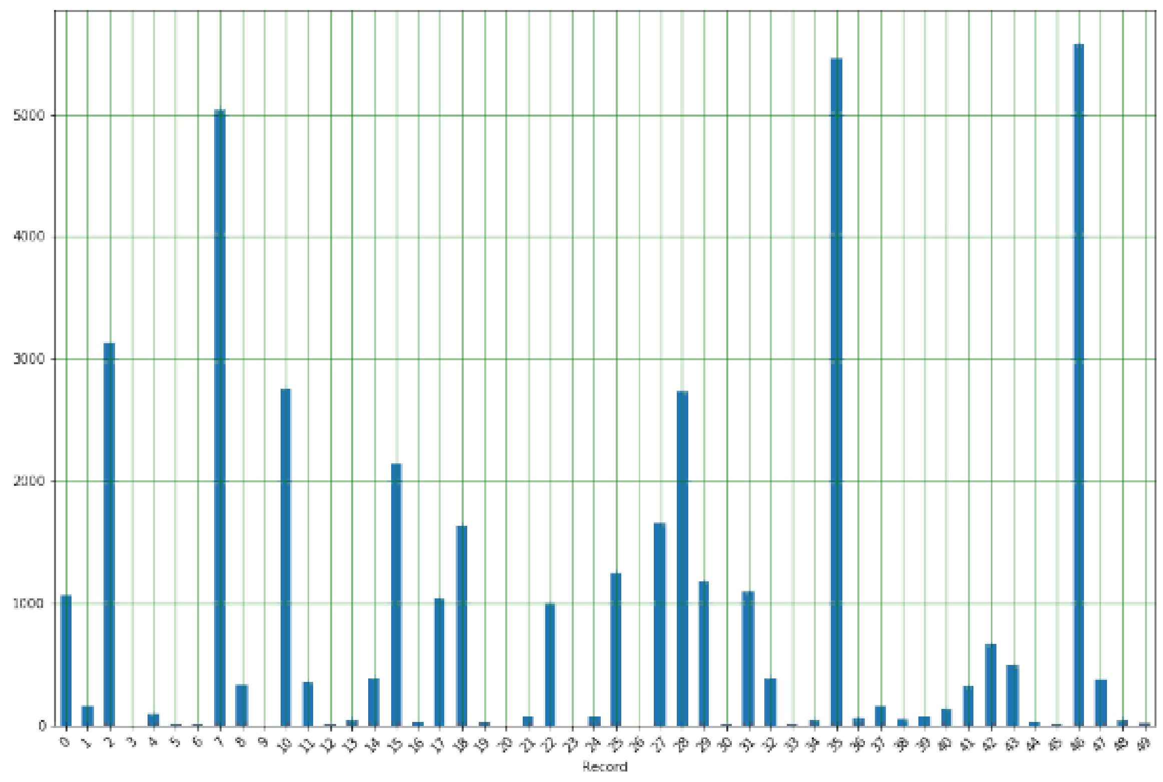


Figure 9 – A bar chart of the first 50 squared errors of the multi-layer perceptron model for each prediction result.

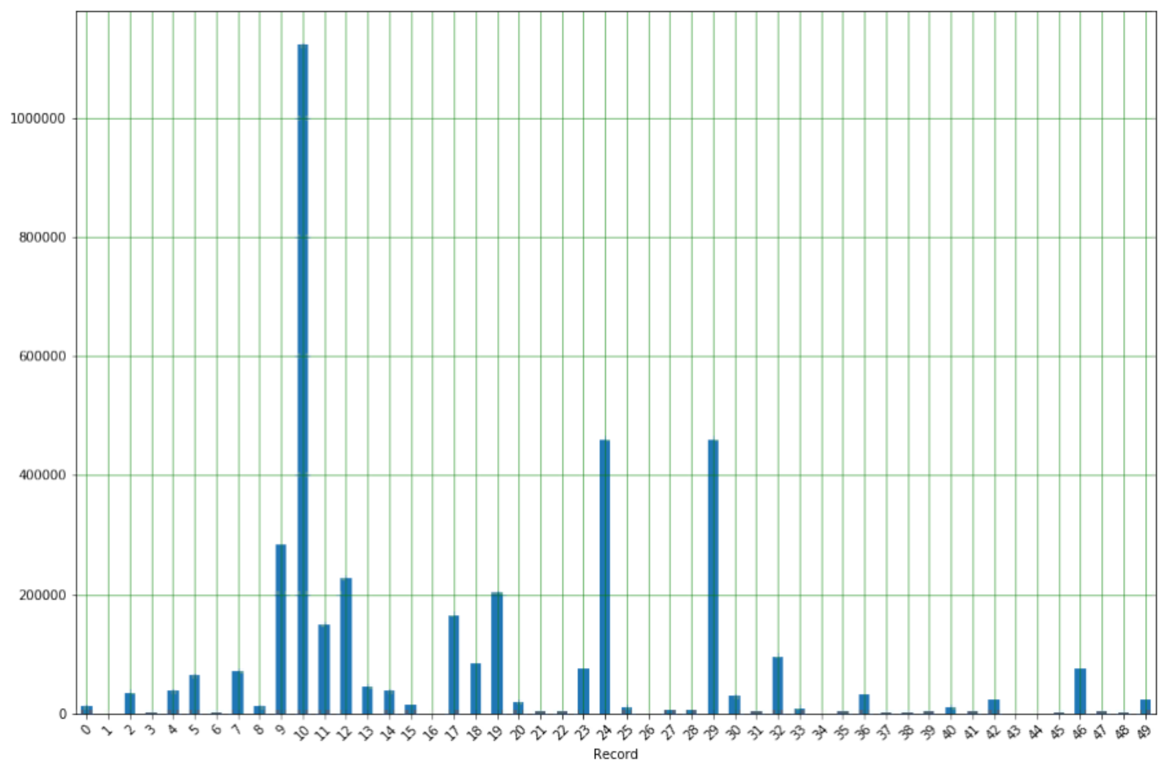


Figure 10 – A bar chart of the first 50 squared errors of the support vector machine model for each prediction result.

Thus, while taking a look at figures 8,9,10 above, it can be realized how significant is the difference between the ranges of each methods' squared errors. So, the narrowest range of errors certainly has a multi-layer perceptron and the widest one has the linear regression model.

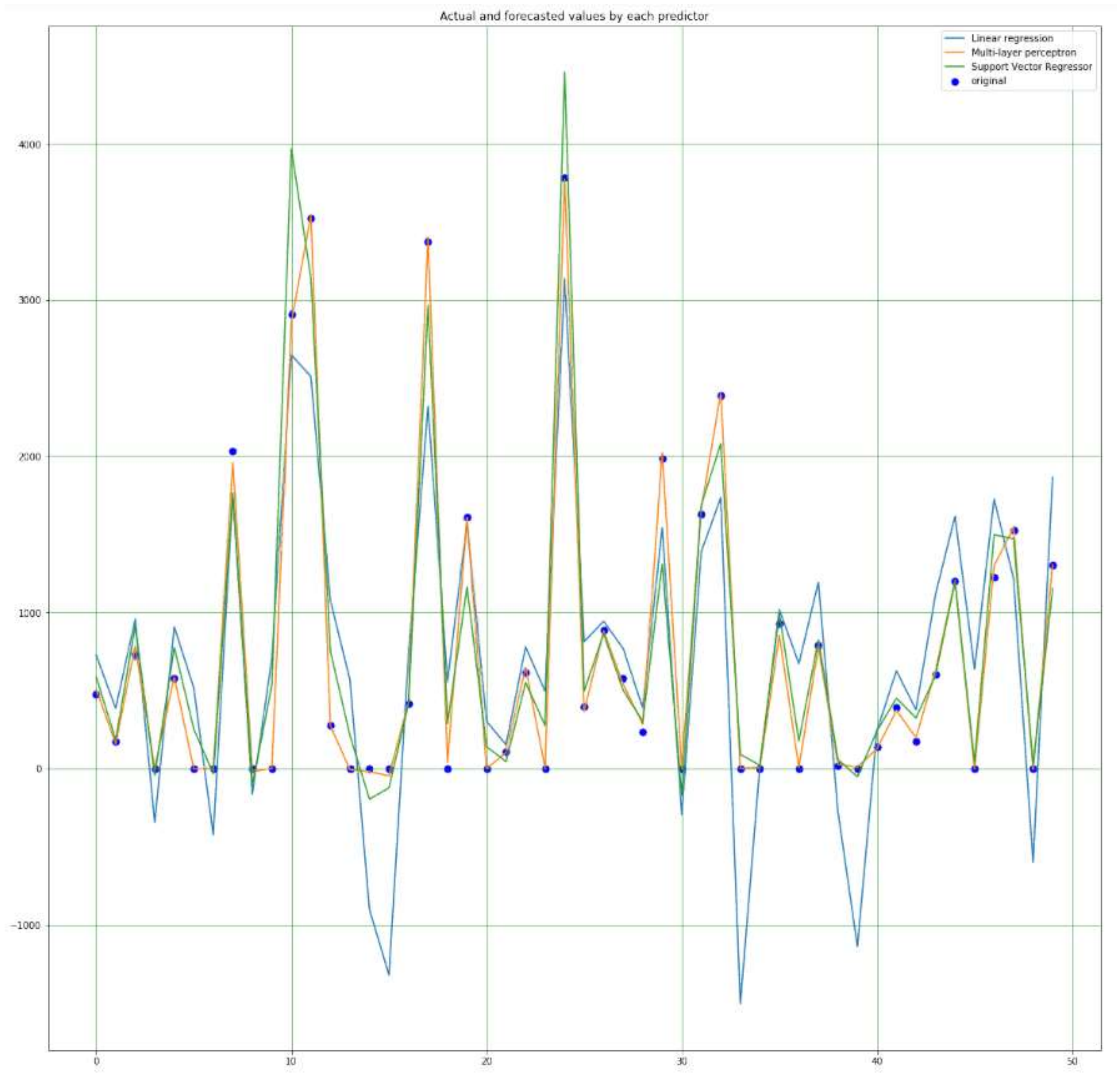


Figure 11 – The comparative plot of actual values from the test data and predicted ones by each method.

In figure 11, it is clearly demonstrated how significant in some spots is the difference between the prediction results of each method. Thus, barely noticeable deviations are inherent in the multi-layer perceptron’s model, while the linear regression model has the most considerable ones. By the way, the support vector machine model in turn has some obvious deviations around the actual values, but they occur rather occasionally.

The prediction results of the final test dataset are demonstrated in the table 11 below.

Table 11 – The table of comparison between predicted values of each model

Predicted #	Multi-layer perceptron	Support Vector Machine	Linear regression
0	502.829691	537.405356	739.920670
1	-4.395854	-99.969252	275.760310
2	3698.208751	3248.694602	2653.437827
3	2470.120800	2138.339931	1765.704389
4	2.854894	-569.030387	151.045057
5	2088.009765	2252.262093	2100.274914
6	226.007456	669.279691	744.634067
7	513.348399	906.738057	842.083932
8	-1.597038	-186.832498	-107.316205
9	2218.319345	1998.867712	1658.471504
10	777.930841	720.911585	890.819016

From the predicted values of the best method (Multi-layer perceptron), the assumption can be made about what the real values are. For example, results -4.395854, 2.854894, and -1.597038 for the customers 1, 4, 8, accordingly, indicate that these customers shall not obtain a discounted premium for the next time. In turn, both the other methods also provide relatively small values for these customers which implies that those people are unlikely to get insured further.

2.3 Summary

According to obtained results from the conducted investigation, we should state that we have managed to predict discounted premium values for each customer and, in particular, determine whether a customer shall get insured or not simply by considering the magnitude of an outcome.

The list of implemented techniques includes linear regression, a support vector machine with a radial basis function kernel, and a multi-layer perceptron. Thus, by applying unique features of each method to solve the given task, all of these classifiers have provided rather fine performance both on train and test datasets.

By the way, we can determine the strengths and weaknesses of applied methods by considering the outcomes

For example, the main strength of linear regression is a rather simple approach that allows it to get adjusted to data even with many variables. On the other hand, this technique can only work properly if the relationship between dependent and independent variables is linear. Otherwise, an application of a linear regression would not have any sense.

Certainly, we did not have a perfect linear relationship between the mentioned types of variables in the considered datasets. However, nevertheless, we have managed to achieve a decent precision for the linear regression model even on such large amounts of data.

Thus, at least on average we may know for sure which customer shall obtain a discounted premium or not. Although, the actual value of a possible discounted premium is rather hard to imagine.

In contrast, an SVM implementation has provided much better performance. Thus, an application of a radial-basis kernel to the SVM model allowed it to achieve 91.2% of accuracy on the test data. Therefore, the main strength of SVM is its excellent performance and applicability to a vast amount of information. However, the weakness of SVM is its sensibility to a scale of data. This fact indicates that a researcher has to investigate how to standardise his raw data in order to obtain a precise model.

According to the obtained results, the SVM model, undoubtedly performs excellently, however, in some cases the outcomes have a considerable bias. Nevertheless, this model still can be used to do its job and demonstrate high accuracy.

The last model – multi-layer perceptron has demonstrated an amazing performance on training data and prepared validation data. In essence, it managed to achieve 99.9% of accuracy and a tiny mean squared error. Considering the obtained values from the test dataset, it is expected greatly to prove its extremely high precision.

Thus, it would be obvious to an insurance manager to determine possibly not-insured customers, because most of the obtained values are grouped around 0.

Concerning the other values, still in some rare cases, this model has a small bias, but perhaps in a real-life, it would not be a rather big difference.

On other hand, this model has many parameters to adjust. Therefore, it would take more time to train on a larger dataset with billions of records. Also, almost always this technique requires data scaling to work properly.

However, the travel insurance company could take advantage of this model to precisely predict the values of discounted premiums for their customer.

All in all, in the result, of the conducted investigation on the ML methods applied to determine a value of a discounted premium, it should be claimed that the given task can be solved using such classifiers as linear regression, a support vector machine with a radial basis function kernel and a multi-layer perceptron. Thus, in essence, a travel insurance company may apply the aforementioned techniques to their needs.